AD-752 754

OPTIMAL WEIGHTED ANCESTRY RELATIONSHIPS

Fred Glover, et al

Texas University

Prepared for:

Office of Naval Research

September 1972

CENTER FOR CYBERNETIC STUDIES

The University of Texas
Austin, Texas 78712

17

Unclassified
Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) Center for Cybernetic Studies The University of Texas	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP

3. REPORT TITLE

Optimal Weighted Ancestry Relationships

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

5. AUTHOR(S) (First name, middle initial, last name)

Fred Glover D. Klingman
T. Klastorin

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
September 1972	~~15~~ 17	7

8a. CONTRACT OR GRANT NO. NR-047-021 b. PROJECT NO. N00014-67-A-0126-0008 c. d. N00014-67-A-0126-0009	9a. ORIGINATOR'S REPORT NUMBER(S) Center for Cybernetic Studies Research Report No. 94 9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)

10. DISTRIBUTION STATEMENT

This document has been approved for public release and sale; its distribution is unlimited.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
	Office of Naval Research (Code 434) Washington, D.C.

13. ABSTRACT

An efficient solution method is given for a class of practical optimization problems requiring the determination of a consistent partial ordering for sets of objects, events, preferences, etc. These problems are characterized by the existence of "noisy" (or contradictory)links of varying strengths. The origin of this class of problems is an anthropological study in which it is desired to specify a global chronological ordering of ancient cemetery data. A mathematical formulation is given which subsumes this and a variety of related problems within the framework of selecting a "maximally consonant" set of arcs in a weighted network. We show that the problem can be solved by an elegant "one-pass" algorithm.

II

DD FORM 1 NOV 65 1473 (PAGE 1)
0101-807-6811

Unclassified
Security Classification

Unclassified
Security Classification

14 KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Optimization problems						
Weighted networks						

III

DD FORM 1 NOV 65 1473 (BACK)
S/N 0102-014-6800

Unclassified
Security Classification

Research Report
CS 94

OPTIMAL WEIGHTED ANCESTRY RELATIONSHIPS

by

Fred Glover*

T. Klastorin**

D. Klingman

SEPTEMBER 1972

*UNIVERSITY OF COLORADO
**WAKE FOREST UNIVERSITY

This research was partly supported by a grant from the Farah Foundation and by ONR Contracts N00014-67-A-0008 and N00014-67-A-0126-0009 with the Center for Cybernetic Studies, The University of Texas.

CENTER FOR CYBERNETIC STUDIES

A. Charnes, Director

Business-Economics Building, 512

The University of Texas

Austin, Texas 78712

I

1.0 Introduction

In this paper we show how to solve a class of optimization problems which has arisen, in a manner of speaking, from the grave. The attempt to specify a global chronological ordering of ancient cemetery data is shown to give rise to a mathematical formulation for a class of problems resembling the traveling salesman problem, but which can be solved by an elegant one-pass, "maximum consonance" algorithm.

The cemetery of the anthropological study which underlies our model consists of a number of individual gravesites each containing a series of artifacts at various ground depths. At each gravesite, pottery types have been identified and marked according to their ground depth. Due to the fact that a grave will sink after a number of years and the site will be reused, it is customary to assume a direct relationship between the age of each pottery type and its ground depth at each gravesite. From such partial (or "relative") chronological orderings at each gravesite, weights may be determined to reflect the relative strength of the ordering in a local sense. (The partial orderings at different gravesites--or even at the same gravesite for different graves--may be inconsistent, and may, thus, involve different "precedence weights" for the same pottery types.) Due to the inconsistency of the data, there is no direct way to use the partial orderings to derive a global partial ordering on the pottery types. Consequently, given these precedence weights, the problem becomes that of finding a "best" partial global ordering of the pottery types according to various criteria.

Viewing each pottery type as a node and each precedence relationship as a directed arc (where a directed arc from node i to node j corresponds to the precedence relationship, i precedes j), our problem is, in a network context, to determine a set of arcs that contains no directed cycles (circuits) and which

optimizes some objective function defined in terms of the arc weights. In particular, the objective of maximizing the sum of the weights on the selected arcs gives rise to the following problem formulation:

$$\text{Maximize} \sum_{(i,j)\in N} w_{ij}\, x_{ij}$$

$$\text{subject to:} \quad \sum_{(i,j)\in \text{Circuit}} x_{ij} \leq \text{one less than \# of arcs in the Circuit for every Circuit.}$$

$$x_{ij} = 0 \text{ or } 1,$$

where N is the set of admissible arcs, w_{ij} is the weight associated with arc (i,j) and x_{ij} is the 0-1 decision variable associated with selecting i to precede j. This problem can alternatively be stated as a weighted covering problem (See Balinski [1]) as follows:

$$\text{Minimize} \sum_{(i,j)\in N} w_{ij}\, y_{ij}$$

$$\text{subject to:} \quad \sum_{(i,j)\in \text{Circuit}} y_{ij} \geq 1, \text{ for every Circuit}$$

$$y_{ij} = 0 \text{ or } 1$$

where $y_{ij} = 1 - x_{ij}$. Unfortunately, however, these formulations require the enumeration of all circuits and thus place the problem at a level of difficulty somewhat beyond that of the ordinary weighted covering problem. In fact, observing that the constraints in either formulation are essentially the "cuts" proposed by Dantzig et al [4] in their approach to the traveling salesman problem, the problem at hand appears to be much closer to the traveling salesman problem than to the weighted covering problem. Moreover, in the absence of enumerating all circuits, this problem has the same definition as the traveling salesman problem except that it allows no master circuit and thus lacks the single additional constraint which requires $\sum_{(i,j)\in N} x_{ij}$ to equal the number of

nodes in the network. In this sense, the problem may be regarded as an "open" traveling salesman problem.

The close relationship of this problem to the traveling salesman problem suggests that efficient solution methods will in general be difficult to come by. Nevertheless, we will show that a natural (indeed, virtually compelling) reformulation of the objective function makes it possible to solve the problem via an elegant one-pass procedure. Our approach is germane to any class of problems which may be characterized roughly as follows: given a collection of data, experiments, historical summaries, etc. that impute weighted partial orderings to certain objects, determine an optimal global ordering of all objects with respect to the weights. In addition, our procedure is useful for testing consistency and isolating inconsistency in the data for such problems.

2.0 Some Ways of Determining Precedence Weights

For many problems, including the gravesite problem, there is a very large number of ways one could determine precedence weights from the raw data. It is not our intent in this section to provide even a partial enumeration of alternative approaches that might be used. However, we shall propose some procedures for imputing weights which have a particularly appealing property. The schemes we propose will be stated in terms of the gravesite problem, but can obviously be applied in other contexts. They all satisfy the condition that $w_{ji} = -w_{ij}$ where w_{ij} is the weight assigned to the precedence relationship, i precedes j.

For the k^{th} gravesite (trial), let p_{ij}^k be a number which reflects the number of times i appears before j. For example, one could increase p_{ij}^k by α_1 if i precedes j by one level, α_2 if i precedes j by two levels, etc. Another possibility would be to reduce p_{ij}^k by an arbitrary amount if the number of times i precedes j is relatively small, thereby implicitly accommodating the fact that such an

ordering might be a freak occurence and deserves less weight. Having determined values for p^k_{ij}, the weights w_{ij} on the arcs (i,j) in the global network can be determined by any one of the following:

1. $w_{ij} = \sum_k p^k_{ij} - \sum_k p^k_{ji}$.

In this instance, w_{ij} represents the net number of times (in a weighted sense) that i precedes j over all trials.

2. $w_{ij} = (\sum_k p^k_{ij} - \sum_k p^k_{ji}) \,/\, [\sum_k (p^k_{ij} + p^k_{ji}) \,/\, n]$

where n is the number of nodes in the network.

Here w_{ij} corresponds to the net number of times i precedes j divided by the average number of times i precedes j or j precedes i.

3. Define the weights w_{ij} as in 2, but let n be any positive real number. (Note that if the objective function is linear in the w_{ij} then 2 and 3 will yield the same optimal solutions.)

In general, for the objective function of maximizing $\sum w_{ij}x_{ij}$, it is reasonable to include in the network only those arcs with positive weights. Further since 1, 2, and 3 will yield w_{ij} values such that $w_{ij} = -w_{ji}$, dropping all zero and negative weight arcs will immediately eliminate the simplest type of inconsistency, and we henceforth assume that such arcs are in fact deleted. The use of the foregoing definitions will avoid inconsistencies in the w_{ij}'s for any particular trial. Thus, it is only by considering two or more trials in conjunction that inconsistencies in the weights can occur.

3.0 Some Objective Functions

There are a few special situations in which the difficulties of solving the problem as formulated in the preceding section can be circumvented. For instance, suppose the data is perfectly consistent after deleting the non-positive arcs. This would mean that if i precedes j, there is no implication that j precedes i

(i.e., if a path exists from i to j, then a path does not exist from j to i). If this "transitivity" condition exists in the network then clearly the circuit constraints are automatically satisfied by every feasible solution. Thus, the problem becomes solvable as an ordinary linear program since the integer restrictions will serve only to restrict the variables to be less than or equal to one. Moreover, in this case, the optimal solution is exceedingly easy to obtain since it simply consists of selecting all positive weight arcs. The solution is also easy to obtain in a few more general cases. For example, if the circuits which are composed only of positive weight arcs are disjoint, it suffices to drop the minimum weight arc in each of these circuits, selecting the positive weight arcs that remain. Similarly, if it is possible to identify a minimum cardinality set of arcs whose removal will break all circuits, and whose sum of weights does not exceed that of any larger cardinality set, then eliminating these arcs will solve the problem. In most cases, however, the problem data is not so amenable to exploitation, and the computational difficulties previously alluded to loom very large indeed.

Disregarding these computational difficulties and taking the liberty of considering alternatives that may more closely reflect the underlying purpose of the model, it would seem especially desirable to modify the problem objective to maximize what might be called the "consonance" of the selected arcs. To accommodate this, it is useful to think of replacing the w_{ij} by "consonance measures" c_{ij}. The concept of consonance can be described by reference to the inequality relationship: If i precedes j ($w_{ij} > 0$) and j precedes k ($w_{jk} > 0$) then $w_{ik} > \max\{w_{ij}, w_{jk}\}$. This sort of transitivity is much stronger than required for the original objective to permit an easily obtained optimal solution, but is what one would expect to encounter if all data were "perfectly

consistent." In a given circuit we can therefore think of the "dissonance" of a given arc as the weight of the maximum weight arc that precedes it (i.e., of all other arcs in the circuit) minus the weight of the given arc. This is the amount by which the given arc conflicts with the inequality relationship if the other arcs are correctly weighted. The consonance of the arc, or the amount of potential agreement of that arc with other arcs in the circuit, can be defined as the negative of the quantity obtained by subtracting the greatest dissonance over the other arcs from its own dissonance. From another standpoint, the minimum weight arc in a circuit is the "most suspect" arc, and consonance can be defined as the weight of the given arc, less the weight of the most suspect (minimum weight) arc on the <u>rest of the circuit</u>. These two definitions give the same rank ordering of consonance in a given circuit, and for circuits of more than two arcs, give the same arithmetic measure of consonance for all arcs except the one of maximum weight. Of course, an arc may be contained in more than one circuit, and we define its consonance over the entire graph (or its global consonance) to be its smallest consonance in any circuit in which it lies. [Note, by either of the definitions of consonance over a circuit this implies that a consonance value over the entire graph is zero or negative if and only if it corresponds to the minimum weight arc in <u>some</u> circuit.]

Our motivation behind this definition of global consonance values is to minimize the amount of dubious orderings that will appear in an optimal solution. To see this, consider the network in Figure 1.

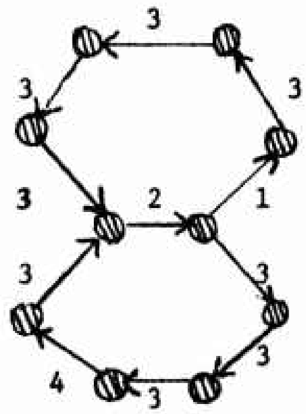

Figure 1 - w_{ij} weights are shown.

Here, the (global) consonance values are negative for the two adjacent arcs of weight 1 and weight 2. Thus, to maximize $\Sigma\, c_{ij}x_{ij}$ (where the c_{ij} are the consonance values), both of these arcs will be dropped, though only the arc of weight 2 "needs" be dropped to eliminate directed cycles. But retaining the "weight 1 arc" is very dubious, since this would imply that it precedes a host of others that should quite possible precede it. Discarding it does not create suspicious relationships, but merely eliminates them. A valid precedence ordering may thus be overlooked, but it becomes less likely that an invalid one will be retained. (Also note that an optimal solution to the maximum consonance problem will be optimal to the maximum weight problem if the network contains no circuits or contains only disjoint circuits.)

4.0 Solving the Maximum Consonance Problem.

There are two main difficulties encountered in replacing the maximum weight problem by the maximum consonance problem. The first difficulty (computationally though not conceptually) is to calculate the consonance values c_{ij}. The second difficulty is that the "form" of the problem has not really changed--we have merely substituted c_{ij} for w_{ij}--and we have already observed that problems of this form are potentially very difficult to solve.

Thus, while maximizing consonance in a network may be more desirable than maximizing "weight", it appears that no gains have been made toward making the problem more tractable. However, it fortunately turns out that the consonance measure is desirable for more than its intuitive appeal. To see this, recall that a consonance value is nonpositive if and only if it is the minimum weight arc in some circuit. Thus every circuit must have at least one arc whose c_{ij} value is nonpositive, and it also follows that the arcs with positive consonance can form no circuits. Moreover, as noted earlier in connection with the maximum weight problem, a problem in which positive weights do not form circuits is one that is very easy to solve. Applied in the present context, this observation leads to the following prescription for solving the maximum consonance problem: Find and eliminate all arcs in the network whose consonance values are negative, together with a sufficient subset of those whose consonance values are zero so that no circuits remain. This nevertheless leaves the initial difficulty of computing consonance values, or of developing a solution algorithm that does this implicitly. We now consider how this may be accomplished.

One approach would be to enumerate all circuits in the network and then to find a minimum weight arc in each circuit. This procedure, while perfectly direct, unfortunately involves as much effort as formulating the maximum weight problem as a weighted covering problem.

Another approach to finding the negative or zero valued consonances would be to order the weights in ascending order and then to consider each arc successively in the network, applying the rule whereby the arc is deleted if it is a minimum weight arc in some circuit of the <u>current</u> network. This approach improves considerably on the preceding but still requires a labeling procedure to determine whether deleting an arc from a graph will break a circuit.

Since the labeling procedure may have to be performed a large number of times, it seems worthwhile to look for better alternatives.

A more effective approach we propose, can be founded on a network construction rather than a network reduction. (We will show subsequently how the details of this construction can be performed in an efficient manner.) To begin, suppose that the weights are arranged in descending order. It suffices then to examine the arcs in sequence, determining whether the arc currently being examined forms a circuit with some subset of the arcs examined previously. If it does, this arc is designated as a "temporary arc" and the process continues. After all arcs have been thus examined, the arcs designated "temporary" are discarded from the network. The arcs remaining provide an optimal solution. The justification of this follows by considering the partial network created by the addition of the current arc to the preceding arcs. Each temporary arc corresponds to the minimum weight arc in a new circuit created in such a partial network. Thus, this arc would have a consonance value of at most zero, and in addition, no circuits can remain once the temporary arcs are removed.

The critical prerequisite to implementing the foregoing algorithm efficiently is the ability to recognize with little effort whether the addition of a new arc creates a circuit in the network being constructed. (A nearly identical problem is encountered by Construction A of Kruskal [6], for identifying a minimum weight spanning tree, but is not resolved in [6], nor in subsequent references such as Berge [2] which describe this method.) We now indicate a simple procedure that will accomplish this task.

Start with an n x n identity matrix I where n is the number of nodes in the network. View the $j^{\underline{th}}$ column of this binary matrix as indicating those nodes which can reach node j (i.e., if the $i^{\underline{th}}$ component of column j is 1 then a directed path exists from node i to node j). Equivalently, we may view the

$i^{\underline{th}}$ row of the matrix as indicating those nodes which can be reached from node i. Note that these interpretations accord with the starting matrix I which corresponds to a network with no arcs. When arc (i,j) is to be added to the network under construction, the updated form of the matrix which accords with the indicated interpretation is created by either of the following equivalent operations:

(1) Replace each column k that has a 1 in row j by the bit union of column i and column k.

(2) Replace each row k that has a 1 in column i by the bit union of row j and row k.

The justification of these operations is established by the following reasoning. Row j contains a unit entry for each node k for which there is a directed path from j to k. Thus, adding arc (i,j) creates directed paths from node i to precisely the set of nodes k so identified. Moreover, for each of these nodes k, the set of all nodes p for which a path is created from p to k (upon adding (i,j)) must be the set of nodes which currently connect by a directed path to node i. This latter set of nodes is given by the unit entries in column i. Consequently the union of columns i and k gives the set of all nodes which connect to node k after adding arc (i,j). This justifies operation (1), and operation (2) is justified similarly.

With the correct form of the updated matrix thus determined, the "temporary" arcs of the algorithm can be identified in a particularly simple fashion. In particular, the arc (i,j) above to be added to the network should be designated temporary if there is a unit entry in cell (j,i) of the matrix. Note that this identification (and its supporting matrix updating) is not only quite simple but is also considerably more efficient than the iterative use of a labeling scheme to detect circuits.

Having thus characterized a one-pass procedure for solving the maximum consonance problem, we can go a step further and specify not only an optimal set of arcs but also the entire partial ordering implied by these arcs. One particularly simple, but efficient, way to obtain this ordering is to shadow the bit union matrix previously discussed with a second bit union matrix. The matrices begin the same, but the updating operations are carried out in the second matrix only when the current arc (i,j) is not designated "temporary." As a result, this second matrix has the same property as the first, but is defined in terms of the optimal solution network instead of the full network. Thus given this matrix, it is quite easy to determine those nodes that node i precedes, those nodes that precede node i and those nodes that neither precede nor follow node i in the optimal solution. Consequently, the bit union matrix approach not only provides a simple solution vehicle but also a convenient way to find and represent the optimal global partial ordering.

5.0 Conclusion and Future Investigations

The concepts and procedures developed in the preceding sections provide a highly efficient one-pass algorithm for sorting pairwise orderings over subsets of data into an optimal partial ordering over the full set. This was accomplished by defining the concept of consonance, and from there devising a procedure to obtain an optimal solution by examining each edge of the network exactly once. A novel feature of this procedure is that the precise values of the consonance measure, which would be extremely laborious to obtain, are never computed--i.e., an optimal solution is generated without explicitly identifying the coefficients of the objective function. This is made possible by analyzing the form an optimal solution must exhibit, and more particularly by characterizing the arcs which must be excluded from this solution (in terms of their

original weights). To provide an efficient means for implementing the algorithm we have specified a convenient approach for identifying the creation of circuits in the process of constructing a network. The approach can also be used in other contexts (e.g., Kruskal [6]) and is far more efficient than the use of a standard labeling procedure. The use of the algorithm also makes it possible to identify consistency difficulties in the weights w_{ij} which the researcher may want to know in order to perform a sensitivity analysis on the ordering.

References

1. Balinski, M.L., "Integer Programming Methods, Uses, Computation," Management Science, Vol. 12, (1965), pp. 253-313.

2. C. Berge, The Theory of Graphs and Its Application, (New York: Wiley,1962).

3. Charnes, A. and W.W. Cooper, Management Models and Industrial Applications of Linear Programming. Vols. I and II. (New York: John Wiley and Sons, Inc., 1961).

4. Dantzig, G.B., Linear Programming and Extensions. (Princeton, N.J.: Princeton University Press, 1963).

5. Edmonds, J., "Matroids and the Greedy Algorithm," Mathematical Programming, Vol. 1, No. 2, (1971), pp. 127-137.

6. J.B. Kruskal, Jr., "On the Shortest Spanning Subtree of Graph and the Traveling Salesman Problem", Proc. Am. Math. Soc. 7, (1956), pp. 48-50.

7. Wagner, H.M., Principles of Operations Research. (Englewood Cliffs, N.J.: Prentice Hall, Inc., 1969).